

ONOS approach to SDN

Architecture & Feature Roadmap

LBNL
July 15, 2015



Architectural Tenets



- High-availability, scalability and performance
 - required to sustain demands of service provider & enterprise networks
- Strong abstractions and simplicity
 - required for development of apps and solutions
- Protocol and device behaviour independence
 - avoid contouring and deformation due to protocol specifics
- Separation of concerns and modularity
 - allow tailoring and customization without speciating the code-base

Avocet Release



- Basis with clearly articulated architecture and clean, modular code mutually supporting each other
- Demonstrated protocol and device independence
- Demonstrated H/A characteristics
- Laid down blueprints for performance & scale
- Started incubation RAFT-based distribution primitives
- Simple, but clean GUI with support for use-cases
- SDN-IP & Packet/Optical use-cases

Blackbird Release



- Optimized subsystems and measured key performance attributes
- Demonstrated that architecture and code exhibit desired scaling characteristics
- Further improved H/A
- Incubated several new subsystems & features
 - modular GUI, RAFT-based stores, IPv6
- Improved and simplified number of existing abstractions
 - flows, intents, distributed primitives, modular GUI

Cardinal Release



- Incubated several new subsystems & features
 - flow objectives, device drivers, application & component config
 - label management, tunnel management, NETCONF providers
 - app security
- Improved and simplified number of existing subsystems
 - modular GUI, multiple GUI views
- Improved distributed core
 - dropped dependency on Hazelcast & added new distributed primitives
 - further improved performance (3->4Mfps, 160->220Kips)
- Developed new use-case demonstrations for ONS
 - CORD, Segment Routing, multi-vendor packet/optical

Drake Release



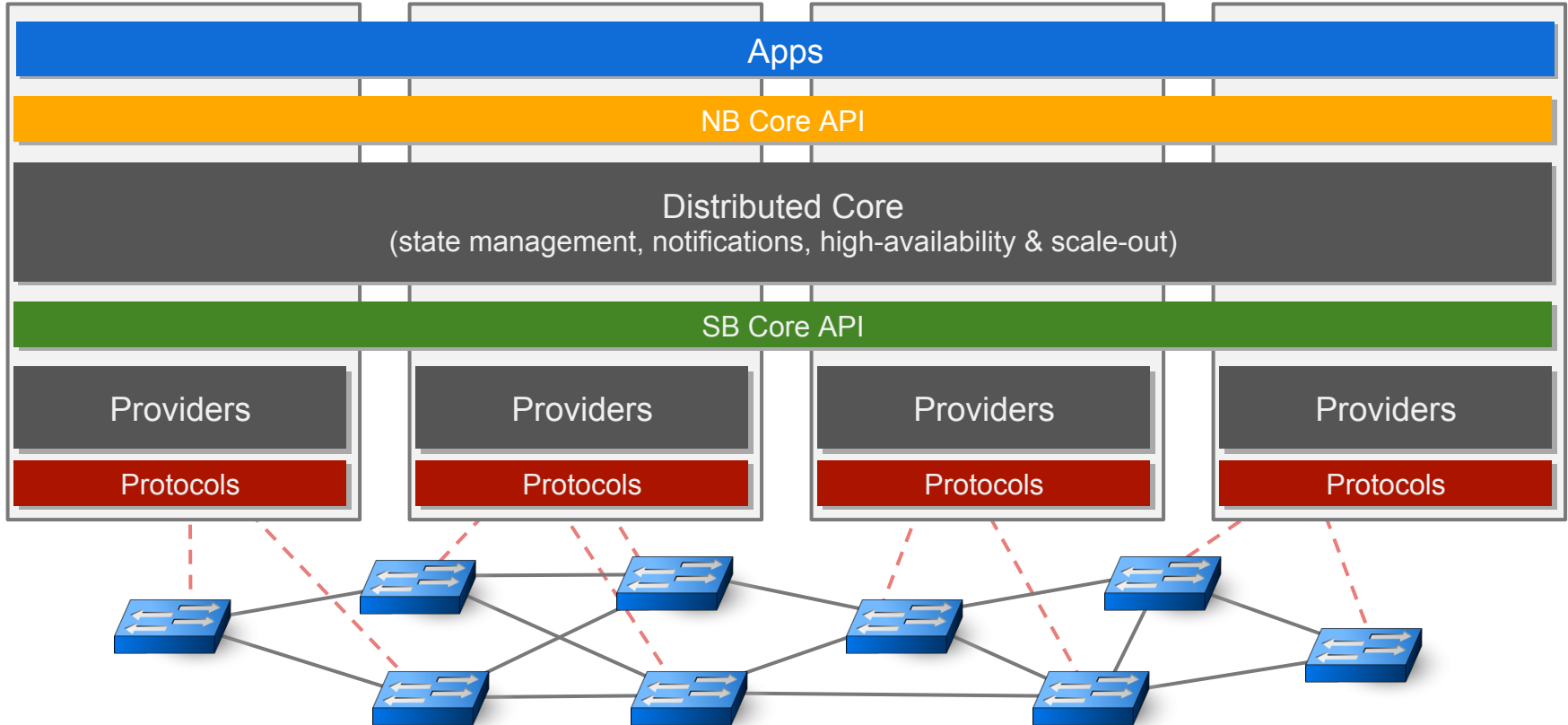
- Platform hardening for larger deployments
 - secure container, REST API & GUI surfaces
 - defect fixes for large networks, intent persistence, REST completeness
 - NETCONF providers, OVSDB providers & additional device drivers
- Incubation of features for OpenStack integration
 - model / API for virtual networks, network functions, multi-tenancy
 - model behaviours for configuring traffic isolation
- Runway items for 2016 use-case trials
 - Scoping & prep for future lab-trials of CORD & Transport SDN
- Complete incubation of features & contributions
 - label & resource management, IPv6
 - GUI topology view overlays & improved traffic visualization

Emu Release

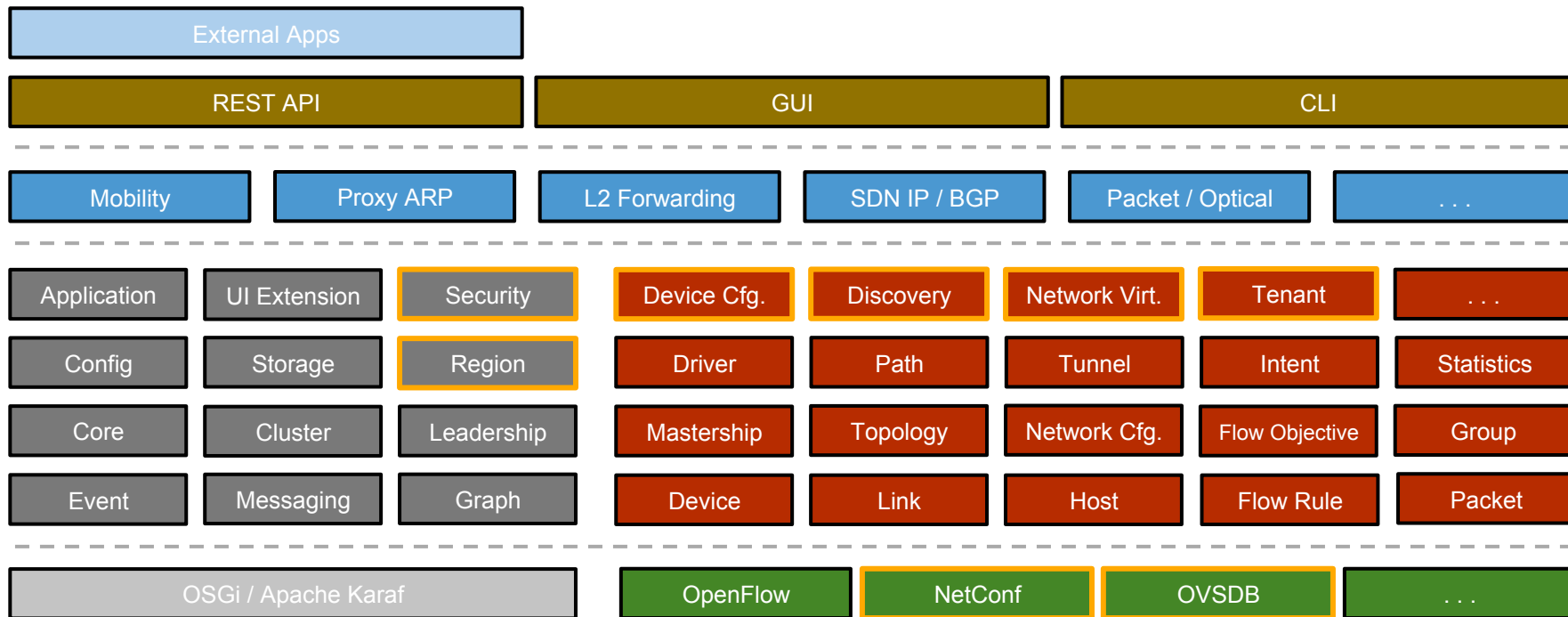


- Complete incubation of features
- Integrate with OpenStack
- Integrate with OPNFV
- Core improvements
 - Dynamic cluster scaling
 - Geo-spread cluster / federation
- Develop new and enhance existing use-cases
 - CORD for lab trials

ONOS Distributed Architecture



ONOS Subsystems - Today & 2015



 Available today

 Roadmap items for 2015

Current Apps & Providers



- Segment routing application
- SDN-IP peering application
- BGP router application
- Proxy ARP/NDP application
- Reactive forwarding application using flow subsystem
- Host mobility application
- A virtual Broadband Network Gateway(BNG) application
- SDN-IP reactive routing application
- ACL application
- Authentication application
- ONOS-XOS integration application
- OLT application
- Packet/Optical use-case application
- Simple fabric application for CORD
- OpenFlow protocol southbound providers
- NetConf protocol southbound providers
- PCEP protocol southbound providers
- Builtin device drivers
- Null southbound providers



Current Deployments

- ONOS / SDN-IP on Internet2 AL2S
- ONOS / SDN-IP on FIU / AmLight
- ONOS / BGP Peering Router at CSIRO
- ONOS / SDN-IP on GEANT, GARR and CREATE-NET
- More to come
 - KREONET in Korea

New Platform Features



- Applications subsystem
 - facilitates easy software deployment across the entire cluster
 - used for apps, drivers and protocol providers
- Component Configuration subsystem
 - facilitates software component configuration across the entire cluster
- Device Drivers subsystem
 - brokerage for device-specific code
 - basis for device configuration
- Flow Objective subsystem
 - allows programming flow rule tables in pipeline-agnostic manner
 - utilizes device drivers

Upcoming Platform Features



- Network Configuration subsystem
 - centralized and extensible means for injecting and tracking meta-data
 - meta-data for instructing the apps and platform on how to interpret and how to act on information discovered about the network
- Network Virtualization subsystem
 - allows creation of networks as slices & splices of underlying networks
 - virtual network behaviours programmed in proactive fashion using intent and flow objective abstractions; not using low-level protocols
 - will support tenant-aware embedded apps
 - tenant specific apps will be required to be external

Upcoming Platform Features



- Intent subsystem enhancements
 - today, intent subsystem allows lateral extensibility; not vertical
 - allow construction of composite intents from various primitives
- Secure Northern and East/West surfaces
 - REST API, GUI & CLI secured using JAAS (via Apache Karaf)
 - southern surfaces excluded for this release
- Initial Device Configuration behaviour models
 - OVSDB & NETCONF based
 - bridge config, port config & queue config

Dev & Test Process Improvements



- Unit test coverage @ 48% and rising
 - assertions of validity is primary objective; coverage metrics secondary
- Scenario smoke tests in Gerrit validation build
 - tests deployment process, fundamental net operations, etc.
 - modular & dev-oriented tests; complementary to TestOn
- Deprecation policy established
 - @Deprecate in V, keep in V+1, remove in V+2
 - incubation area & @Beta entities are exempt
- Incubation Area
 - provides home to subsystems and features destined for ONOS core
 - encourage experimentation, innovation and early exposure

Distributables



- General purpose run-time
 - `wget` http://downloads.onosproject.org/release/onos-*.tar.gz
- Docker run-time
 - `docker pull onosproject/onos`
- AMI - Dev & Test Cell
 - ONOS Cluster & Mininet for dev & test
- ONOS run-time Linux packages
 - `.deb` & `.rpm` packages will follow in Drake/Emu

ONOS & Approach to SDN



- Move with urgency, but deliberately
- Mind the fundamentals & beware of yak-shaving
- Keep balance between innovation, utility and stability
- Allow *legacy* devices to participate in SDN, but not to deform or diminish the SDN vision



Software Defined Transformation of Service Provider Networks

Join the journey @ onosproject.org